PAC-MEN: Personal Autonomic Computing Monitoring Environment

Roy Sterritt
School of Computing and Mathematics,
University of Ulster
Northern Ireland
r.sterritt@ulster.ac.uk

David F. Bantz
IBM T.J. Watson Research Center,
Yorktown Heights, New York 10598
USA
bantz@watson.ibm.com

Abstract

The overall goal of this research is to improve the 'environment awareness' aspect of personal autonomic computing. Personal Computing offers unique challenges for self-management due to its multi-equipment, multi-situation, and multi-user nature. The aim is to develop a support architecture for multi-platform working, based on autonomic computing concepts and techniques. Of particular interest is collaboration among personal systems to take a shared responsibility for environment awareness. Concepts mirroring human mechanisms, such as 'reflex reactions' and the use of 'vital signs' to assess operational health, are used in designing and implementing the personal computing architecture. A proof of concept self-healing tool is considered and lessons learned used for the requirements specification of the community-based environment awareness prototype environment—PAC-MEN (Personal Autonomic Computing Monitor ENvironment).

1. Introduction

Personal Autonomic Computing is Autonomic Computing [1] in a personal computing environment [2]. Personal computing has evolved substantially in the last few years. Its scope now extends from end user computing in the office, to home PCs, wireless laptops, palm tops and next generation mobile phones. In the near future these will be leaf nodes in the self-managing grid infrastructures and next generation internet being developed to deliver eApplications such as eHealth, eScience and eGovernment.

Personal computing is an area that can benefit substantially from autonomic principles. Examples of current difficult experiences that can be overcome by such an approach include [2]: (i) trouble connecting to a wired or a wireless network at a conference, hotel or other work location; (ii) switching between home and work; (iii) losing a working connection (and shouting across the office to see if anyone else has had the same problem!); (iv) going into the IP settings area in Windows and being unsure about the correct values to use; (v) having a PC which stops booting and needs major repair or re-installation of the operating system; (vi) recovering from a hard-disk crash; and (vii) migrating efficiently to a new PC. Coping with these situations should be routine and straightforward but in practice such incidents are typically stressful and often waste a considerable amount of productive time.

Personal computing also creates some problems for the implementation of autonomic principles. In particular [2], personal computing users are often, of necessity, system administrators for the equipment they use. Most are amateurs without formal training, who perform system operations infrequently. This reduces their effectiveness and typically requires them to consult with others to resolve difficulties.

The PAC-MEN project is involved with researching, designing and developing a personal computing autonomic framework that addresses the type of problem identified above. In particular, it will use the notion of a consultation between each personal system and a network of dynamically discovered peers to resolve difficulties experienced [2][3], addressing such questions as how to connect to a new network or locate local resources. This peer model may be extended to include shared monitoring of the external environment to inform group members of events that may require individual action. An obvious example is the detection of a virus or other malicious attack.

To clarify the basic requirements and activities of the general monitoring autonomic system architecture for creating a supportive personal autonomic computing environment the paper considers related work and the developed proof of concept [4]. From lessons learnt the paper then describes the further planned development.

2 Related Work

Although the autonomic computing area is relatively new it is attracting widespread attention both from industry and academia. In particular there are significant self-managing system industry initiatives as well as from IBM (Autonomic Computing); HP (Adaptive Infrastructure), Sun (N1), Cisco (Adaptive Network Care) and Microsoft (Dynamic Systems Initiative). Examples of academic involvement include work at

Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04)
1529-4188/04 $ 20.00 IEEE

Rutgers (active middleware) [5], CMU (self-healing systems) [6], Columbia (retrofitting legacy systems) [7], and Imperial College (autonomic management of ubiquitous eHealth systems [8] and autonomic web [9]).

Most of this work, especially that in industry, has focused on server management [3], since requirements for reliability have increased, while servers have become more complex and hence more difficult to maintain. The contribution of autonomic computing to personal computing is different—being much less about achieving optimum performance or exploiting redundancy and more about simplifying use of the equipment and the associated services involved [3]. Within autonomic personal computing, examples of current research include: (i) automatic system configuration, especially for application or communication settings, as in Prism, which uses a rule-based approach (ABLE) [10]; (ii) sharing workloads through a PC-based local Grid [11]; and (iii) autonomic management of wearable health devices [8].

Peer frameworks are becoming mainstream, for instance JXTA [12] and Microsoft's peer framework update [13]. The paradigm is also key in ambitious future plans for virtual file servers that would be accessible to a hundred thousand [14] or even link billions [15] of individual computers. The peer-to-peer paradigm offers the flexibility required for achieving autonomic personal computing.

3. PAC-MEN

The main objectives of the PAC-MEN research is to;

(a) Define a support architecture for multi-platform working, based on autonomic computing concepts and techniques. Particular concepts of interest are (i) collaborative monitoring across personal systems; and (ii) the mirroring of human mechanisms such as 'reflex reactions' to respond to threats, and the use of 'vital signs' to assess operational health.

(b) Provide personal computing support across a range of platforms from PCs, mobile laptops, PDAs, to wearable devices, involving a range of connection paradigms such as virtual, peer-to-peer, client-server and grid.

1. Personal Autonomic Computing Requirements

From a user perspective, the broad requirements for personal autonomic computing are for improved usability and security, without any significant performance degradation. Performance may be an issue under current popular OS, as each autonomic manager will be repeatedly executing a 'monitor, analyse, plan and execute (MAPE)' loop [16] to identify and respond to problematic situations. Most time will be spent in monitoring and analysing significant events that occur, the cost of which will obviously depend on their frequency of occurrence. Costs can be reduced through local correlation of event messages [17] and by sharing monitoring responsibilities across a group of collaborating personal systems.

One helpful characteristic of personal systems is that the users involved are potentially a rich source of system knowledge. It is therefore convenient to make use of that knowledge when recorded information proves inadequate. In effect, a collaboration at both system and user level.

Unfortunately users can also inhibit autonomic operations. For instance, if their machine is to be used collaboratively in the local network all disconnections from the network must be performed in a controlled way. This implies that autonomic activity on a personal device should be visible to the user and explained as necessary [18].

2. Personal Autonomic Computing Architecture

Achieving high usability and security for personal systems requires rapid responses to changing circumstances. The PAC architecture incorporates a mechanism equivalent to the biological reflex reactions, to alert members of the peer group to situations requiring urgent attention. Human reflex reactions enable a rapid response to pain, such as when a hot object is touched. In computing terms, it is assumed that a system will have to reconfigure itself to avoid a detected threat, while maintaining its operation as far as possible. This may result in the system operating with a reduced set of resources [19]. Like the body, a system can then address the problem causing the reaction with less urgency; this may involve some damage repair.

Figure 1 shows an abstract view of a system architecture to support this model [20]. This is similar in nature to the architecture proposed in the IBM blueprint where an autonomic manager consists of monitor, analyse, plan and execute (MAPE) components [16].

An autonomic element is made up of a managed component and an autonomic manager. The self-monitor actively observes the state of the component and its external environment, drawing conclusions using information in the system knowledge base. If necessary, this can lead to adjustments to the managed component. One additional feature is the use of a heartbeat monitor (HBM) extended to a pulse monitor (PBM) to summarize the state of the managed component for other connected autonomic elements. Essentially it provides an indication of the health of the managed component or external environment as viewed by that manager, with the absence of a signal (heartbeat) indicating a specific problem with the manger itself. The signal itself, like a pulse, can provide additional information to further explain the state of the element and trigger reflex actions.

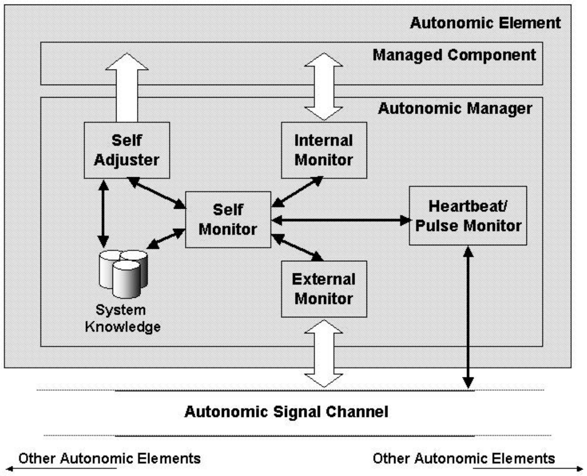

Figure 1 Architecture of an Autonomic Element [20]

NASA has a similar construct, the *Beacon monitor* [21]. A spacecraft sends a signal to the ground that indicates how urgent it is to track it for telemetry data (the beacon states are *nominal*, *interesting*, *important*, *urgent*, and *no tone*). This concept involved a paradigm shift for NASA from routine telemetry downlink and ground analysis to onboard health determination and autonomous data summarisation.

The research reported in this paper is investigating the effectiveness of this architecture and seeking the best form of relationship between the pulse monitor and the external monitor. Conditions and techniques for inducing a change in pulse due to changing conditions in the environment are determined by rules and policies that may self-adapt over time.

4. Vital Signs & Pulse Monitor Prototype

A personal autonomic computing self-healing tool has been designed as an initial proof of concept [4]. The assumption behind the tool is that dying/hanging processes on a PC are signs or indicators of the health of that PC. These *vital signs* may indicate that the PC is becoming unstable and possibly in immanent danger of hanging or unreliable for current processes running on that machine. Peers are notified of this situation via a change in *pulse*.

This is particularly useful in situations where the PC is unattended for example a web server, and the user may be notified via a peer PC that the machine is in difficulty. Another useful situation is when machines in the peer group are sharing work load, for example via Harmony PC grid services [11]; a peer is notified in advance of immanent danger and can recover data and relocate work to another peer. Such an approach is more proactive than responding once the machine has hung, and as such offers fuller potential for autonomic capabilities.

The underlying functionality of the tool is a heart-beat monitor; if a process hangs it should be restarted and the pulse monitor takes note. Upon several processes hanging or the same process repeatedly hanging within specified timeframes, a change occurs in the monitor's perception of how healthy the machine is and as such brings about a change in the pulse being broadcast from that PC.

Since the tool operates in a P2P mode it also takes responsibility to watch out for its neighbours; as such other PCs (peers) will register with it and it will monitor their pulse.

Figure 2 shows the overview of the Pulse Monitoring construct. An internal monitor inside a host takes care of monitoring its health condition which is represented by a Pulse. Each host is able to send its Pulse to a peer via an external monitor. The 'knowledge & database' stores the pulse level and rules (i.e. predefined knowledge) which may adapt over time; the monitoring logs; and the history of neighbour hosts. A computer system is different from a biological system; human biology reflection is involuntary while the decision making in computer system is based on a set of predefined rules or policies. For example, rules such as the Pulse sending interval and terminate the failed process after three trials of re-starting the process, are re-configurable.

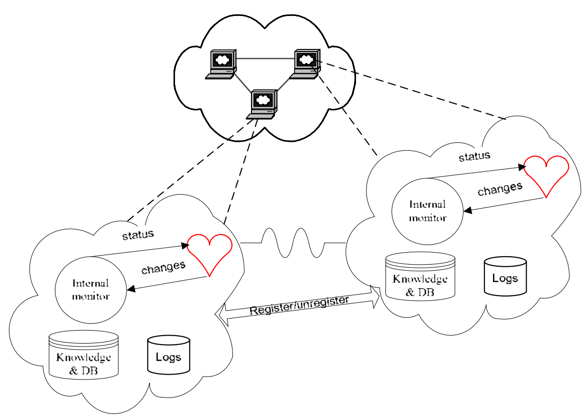

Figure 2 – Pulse Monitoring [4]

The connection between two hosts is established using UDP (User Datagram Protocol). The host sends the degree of urgency to the peer's pulse external monitor instead of just a 'beat'. The urgency level is transformed based on the number of fail process:

The amount of processes required to cause a change in pulse is adaptable and need not necessarily remain at the values depicted in Table 1, as is the time window for qualifying failing processes.

Table 1 – Pulse value

Level	Description	Pulse Change Trigger (adaptable)
0	Nominal	no failed process
1	Interesting	1 failed non-essential process
2	Important	1 failed essential process *or* 2 failed non-essential processes
3	Urgent	2+ failed essential processes or 3+ failed non-essential processes
—	No Pulse	Pulse monitor, or comms has failed

5. PAC-MEN Planned Development

From the proof of concept implementation of the pulse monitor (a reflex reaction with different levels of urgency) and vital signs (dying processes), the PAC-MEN system will further evaluate and refine the personal autonomic computing research and architecture. The proof of concept focused on internal monitoring of a managed component with a single vital sign (failing processes). Using multiple vital signs, for example to also include performance metrics as an indicator used to bring about a change in pulse, would provide a more robust solution.

The planned development of PAC-MEN will be to focus on the external monitoring; creating shared dynamic group environment awareness. In this scenario the pulse becomes a shared value of environment health as opposed to an individual's health value. It should be considered in addition to the self-awareness pulse mechanism. Three key aspects are planned as summarised below.

PAC-MEN-1: The reflex pulse mechanism can be used in any configuration, such as the grid [22] and in a telecom fault management architecture [23]. These particular environments are hierarchical. The proposed peer-to-peer (P2P) approach in this project will require a more distributed solution since no single element has exclusive responsibility for management. In principle, this is an advantage as it shares the burden of environmental awareness among the inter-connected personal computing systems and can improve robustness through redundancy.

PAC-MEN-1 is intended to assess the basic communications architecture. Research issues to be addressed in this phase include: (i) the exact means of collaborative communication (e.g. gossip protocol), and (ii) the analysis and response to the information received.

PAC-MEN-2: For PAC-MEN-2, the development will focus on the shared co-operative monitoring mechanisms. Research challenges that will be addressed in this phase include: (i) establishing the rules and policies for allocating monitoring duties; and (ii) determining how to react to the connection and disconnection of personal devices in the network.

Each machine (the managed component) must have its own internal monitoring but it is also clear that each autonomic manager needs to monitor the external environment. In a peer-to-peer network an individual machine or group of machines (e.g. the least busy PC or assigned PCs) may take on the monitoring role for the group. Other machines must be ready to take on monitoring responsibilities should circumstances change or react to an environment situation indicated by a change in the environment pulse—a reflex reaction.

To facilitate robustness multiple approaches for monitoring and probing should be utilised in a dynamically activated mechanism for example to monitor the environment differently under undesirable circumstances. As such different pulse levels – like urgency levels - would trigger different monitoring mechanisms within the autonomic system.

Ideally this would include extending the architecture with a reflection level to monitor and adjust strategies dynamically [24].

PAC-MEN-3: PAC-MEN-3 will extend the evaluation to investigating environments in which the monitoring is noisy and/or optional, and investigate scaling to large numbers of systems.

6. Discussion and Conclusion

Overall, autonomic computing is intended to improve the general usability and manageability of computing systems and so, in principle, will benefit all computer users in due course. Since for the majority of users access to computing is through personal devices autonomic research in this area should have a significant impact. In the longer term, the work is of direct relevance to emerging important areas, such as utility/grid and ubiquitous computing, which require systems to self-manage to fulfil their potential. These will provide broad support for eScience, eGovernment, eHealth, and eBusiness applications which for the foreseeable future will be accessed by the majority of users through personal computing.

The research in this paper explores a novel computing structure for the distributed realization of autonomic behaviour in personal and embedded systems. Autonomic computing behaviour is commonly implemented in an "Autonomic Manager" (AM), most often a component of a managed system, but sometimes separated from that managed system in a management server. The flow of monitoring and management is most often hierarchical. This flow is appropriate for systems of static or near-static structure, with adequate resources to devote to the AM and to communications between it and the managed system. The contribution in this research is a way for AMs to share data and management decisions in a non-hierarchical way, even in an ad hoc manner. It permits one AM to monitor the health of other AMs without necessarily controlling them, and even to reach decisions based on consensus. It opens up opportunities for the collaboration of AMs in a way that is less rigid than in current autonomic computing architectures. This style of autonomic computing is much more appropriate to personal and embedded computing, because it supports the dynamic, self-centred style exhibited by this type of computing.

The "reflex reaction" notion is based on the need for a system to respond more quickly than it can respond after detailed analysis. It has been observed that failures in recovery from virus infections, where the OS is reinstalled, is re-infected during the installation process itself. Recovery actions when connected to a network hosting a virulent virus infection have to be performed in less time than it takes to get infected. Also, since comprehensive analysis in preparation for response may involve contacting neighbours and servers and waiting in their queues, the time to do this comprehensive analysis may be long. To address this requires research into the dynamics of infection, analysis and recovery so as to supply bounds on the amount of time that a system can take before it starts to take action.

PAC-MEN (Personal Autonomic Computing Monitor ENvironment)—will be further designed, developed, tested, and refined to further assess the concepts and mechanisms proposed in this paper. The success of this research would extend the realization of autonomic behaviour to groups of computers which organize themselves to realize shared objectives. This is a vital step in scaling autonomic computing.

Acknowledgements

The development of this paper was supported by the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

References

[1] Horn P, "Autonomic computing: IBM perspective on the state of information technology", IBM October 2001

[2] Bantz DF, Bisdikian C, Challener D, Karidis JP, Mastrianni S, Mohindra A, Shea DG, Vanover M, "Autonomic personal computing", IBM Systems Journal, 42(1), 2003, pp. 165-176

[3] Bantz DF, Frank D, "Challenges in Autonomic Personal Computing, with Some New Results in Automatic Configuration Management", Proc. IEEE INDIN; AUCOPA workshop, Banff, Alberta, Canada, 22-23 August 2003

[4] Sterritt R, Chung S, "Personal Autonomic Computing Self-Healing Tool" Proceedings of 11th IEEE ECBS - Engineering of Autonomic Systems Workshop (EASe'04), Brno, Czech Republic, May 24-27, 2004

[5] Bhat V, Parashar M, "Discover Middleware Substrate for Integrating Services on the Grid," Proc. 10th Int. Conf. High Performance Computing, Hyderabad, India, Dec. 2003

[6] Garlan D, Cheng S, and Schmerl B, "Increasing System Dependability through Architecture-based Self-repair", in Architecting Dependable Systems, Springer-Verlag, 2003.

[7] Kaiser G, Gross P, Kc G, Parekh J, Valetto G, "An Approach to Autonomizing Legacy Systems" Workshop on Self-Healing, Adaptive and Self-MANaged Systems (SHAMAN), June 2002.

[8] E. Lupu, M. Sloman, N. Dulay, J. Sventek, AMUSE: Autonomic Management of Ubiquitous Systems for e-Health, EPSRC project, http://www.doc.ic.ac.uk/~ecl1/projects/AMUSE/, 2003.

[9] McCann J, Jawaheer G, "The Patia Autonomic Webserver: Feasibility Experimentation", Proc. IEEE DEXA Workshops - ACS, Prague, Czech Rep., Sept. 1-5, 2003, pp 661-667

[10] Bigus JP et.al., "ABLE: a toolkit for building multiagent autonomic systems," *IBM Systems J.,* 41(3), pp.350-371, 2002

[11] Naik VK, Sivasubramanian S, Bantz DF, "Harmony: A Desktop Grid for Delivering Enterprise Computations," Proc. 4th Int. Workshop on Grid Computing, Phoenix, Arizona, Nov. 2003.

[12] Project JTXA, http://www.jxta.org/

[13] Microsoft, Peer-to-Peer Infrasture, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/p2psdk/p2p/portal.asp

[14] Microsoft Research, Farsite project, http://research.microsoft.com/sn/Farsite/

[15] UC-Berkeley, OceanStore Project, http://oceanstore.cs.berkeley.edu/

[16] IBM, "An architectural blueprint for autonomic computing", April 2003.

[17] Sterritt R, "Towards Autonomic Computing: Effective Event Management", Proc. 27th Annual IEEE/NASA Software Engineering Workshop, Greenbelt, MD, Dec. 2002,. pp 40-47.

[18] Russell DM, Maglio PP, Dordick R, Neti C, "Dealing with ghosts: Managing the user experience of autonomic computing", IBM Systems Journal, 42(1), 2003, pp. 177-188

[19] Bapty T, Neema S, Nordstorm S, Shetty S, Vashishtha D, Overdorf J, Sheldon P, "Modeling and Generation Tools for Large-Scale, Real-Time Embedded Systems", Proc 10th IEEE ECBS, Huntsville, Alabama, USA, 7-10th April 2003, pp11-16.

[20] Sterritt R, Bustard DW, "Towards an Autonomic Computing Environment", Proc. IEEE DEXA 2003 Workshops - 1st Int. Workshop on Autonomic Computing Systems, Prague, Czech Republic, September 1-5, 2003, Pages 694-698.

[21] Wyatt J, Hotz H, Sherwood R, Szijjaro J, Sue M, "Beacon Monitor Operations on the Deep Space One Mission", 5th Int. Sym. AI, Robotics and Automation in Space, Tokyo, Japan, 1998

[22] Sterritt R, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID", Proc. IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003, Banff, Alberta, Canada, 22-23 August 2003

[23] Sterritt R, Gunning D, Meban A, Henning P, "Exploring Autonomic Options in an Unified Fault Management Architecture through Reflex Reactions via Pulse Monitoring" Proc Proceedings of 11th IEEE ECBS - Engineering of Autonomic Systems Workshop (EASe'04), Brno, Czech Republic, May 24-27, 2004

[24] Norman DA, Ortony A, Russell DM, "Affect and machine design: Lessons for the development of autonomous machines", IBM Systems Journal, Vol 42, No 1, 2003, pp. 38-44